Machine Learning Nanodegree
Capstone Project Proposal

Named Entity Recognition

Nitin Bhandari
Feb 23, 2018

1 Domain Background

"Named Entity Recognition (NER) also known as entity identification, entity chunking, and entity extraction is an important sub task of information extraction that locates and classifies named entities in text into pre-defined categories such as name of the person, location, expressions of time, monetary values, etc." [1]. It helps to identify the context in which it is written to make sense and extract valuable information from it. But NER possesses serious issues itself. First, there are a huge number of languages and we need annotated data for machines to learn from them. So, quality and quantity of data are one of the key factors to decide the performance of the system. The named entity recognition is a one of the early steps before we can have computers to make sense of the language we speak and write.

For example:

Elon deleted SpaceX page on Facebook on March 23, 2018.

And the system produces an annotated block of text that shows the name of entities:

[Elon]$_{\text{Person}}$ deleted [SpaceX]$_{\text{Organization}}$ page on [Facebook]$_{\text{Organization}}$ on [March 23, 2018] $_{\text{Time}}$

1.1 Motivation

The first time I was introduced to machine learning, I learnt about Natural Language Processing and did an introductory project on named entity recognition using Naïve Bayes. So, for the capstone project, I choose the same domain to build an efficient system.

2 Problem Statement

NER is a subtask of information extraction. While the expression 'named entity' may seem the task to those entities such as word or phrases, stands consistently for a particular reference like those designated as rigid designators but is in fact not strict and can be used for temporal expressions.

For example:

In sentence ‘The automotive company created by Henry Ford in 1903’, Ford refers to the Ford Motor Company although there are other meanings of Ford as well.

‘The year 2001’ and ‘I take my vacations in June’, the first phrase refers to 2001 year of the Georgian calendar and the second phrase refers to the month June of unknown year.

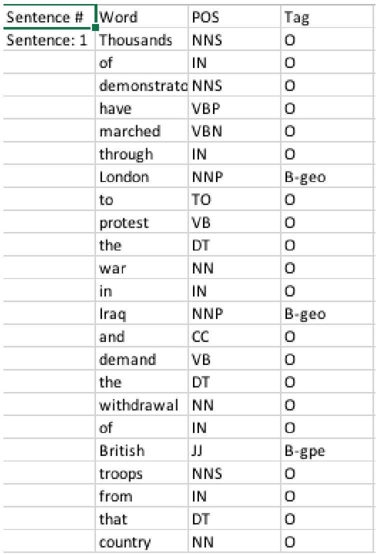
The problem of NER is often broken down into two distinct problems:

a) Detection of named entities (segmentation) and
b) Classification of the name by the type of entity they refer to (Ontology)

The input to the system is the .csv file in which each word in the sentence is annotated with part of speech (POS) tags as shown in figure 1. We use cross-validation and tune the hyper parameters and finally tested on test data and the performance is calculated my looking at the if the system is able to produce the same annotations as given for test data. The output should be the accurately identified tags corresponding to the words in the sentence.

3 Dataset and Input

The NLTK python library does not have a standard corpus for NER in English. But it does have the famous CoNLL 2002 Named Entity CoNLL but in Spanish and Dutch. We will be using dataset provided on the Kaggle annotated with IOB and POS tags [4]. The dataset can be downloaded from here.



Sentence #	Word	POS	Tag
Sentence: 1	Thousands	NNS	O
	of	IN	O
	demonstrato	NNS	O
	have	VBP	O
	marched	VBN	O
	through	IN	O
	London	NNP	B-geo
	to	TO	O
	protest	VB	O
	the	DT	O
	war	NN	O
	in	IN	O
	Iraq	NNP	B-geo
	and	CC	O
	demand	VB	O
	the	DT	O
	withdrawal	NN	O
	of	IN	O
	British	JJ	B-gpe
	troops	NNS	O
	from	IN	O
	that	DT	O
	country	NN	O

Fig 1: Sample Input with tags

Following are the essential information about the entities:

- geo = Geographical Entity
- org = Organization
- per = Person
- gpe = Geopolitical Entity
- tim = Time Indicator
- art = Artifact
- eve = Event
- nat = Natural Phenomenon

The sentences in the text has been divided into words and each word is annotated. We will be splitting the data into cross-validation and test set.

4 Solution Statement

The model relies heavily on good annotated corpus for the system to learn and produce great results. We are going to implement a basic Conditional Random Field (CRF) system for NER and modify the parameters to get a better result.

5 Benchmark Model

Perceptron classifier:

We choose perceptron classifier for NER. The reason being to see how well CRF model performs as compared to linear model (perceptron).

CRF Classifier:

The CRF being a class of statistical modelling method which is used in pattern recognition and often used for parsing sequential data in Natural Language Processing(NLP).

A well-designed CRF model should outperform the perceptron model but linear models also tend to perform well on such data.

6 Evaluation Metrics

The project will be using F1 score to check the performance of the system.

7 Project Design

7.1 Programming Language and Libraries

The project requires:

- Language : Python
- Libraries : pandas, numPy, sklearn, eli5

7.2 Machine Learning Design

7.2.1 Load the data into DataFrame using pandas
7.2.2 Retrieve sentences along with their labels
7.2.3 Prepare the dataset
7.2.4 Implement CRF using sklearn-crfsuite
7.2.5 Perform cross-validation
7.2.6 Get the precision, recall and F1 score using scikit-learn classification report
7.2.7 Use elif5 library to visualize transition probabilities from one tag to another
7.2.8 Parameter Tuning
7.2.9 Go to step 6 and repeat step 8 and step 6 till you achieve a higher accuracy

References:

[1] https://en.wikipedia.org/wiki/Named-entity_recognition
[2] https://nlpforhackers.io/named-entity-extraction
[3] http://www.iro.umontreal.ca/~lisa/pointeurs/RNNSpokenLanguage2013.pdf
[4] https://www.depends-on-the-definition.com/named-entity-recognition-conditional-random-fields-python
[5] https://en.wikipedia.org/wiki/Conditional_random_field